

Tencent Hub

最佳实践

产品文档



腾讯云

【版权声明】

©2013-2020 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

文档目录

最佳实践

Java 自动构建 workflow

容器服务 CICD workflow

最佳实践

Java 自动构建 workflow

最近更新时间：2019-09-11 10:56:09

本文主要叙述如何实践一条 Java 自动构建 workflow。

操作页面

在项目仓库页面->workflow->新建 workflow。

实践步骤

1. 新建 workflow，填写 workflow 基础设置信息。如还未绑定源代码授权，请参考 [源代码授权设置](#)。

← 新建 workflow workflow 模版 环境变量设置 ⓘ 创建工作流指引

GitServer地址 (选填) ⓘ
https://gitee.com

Organization
zhongfox

Git仓库地址
maven_demo

触发方式

满足以下任意条件即触发构建镜像

添加新Tag时触发

提交代码到分支时触发

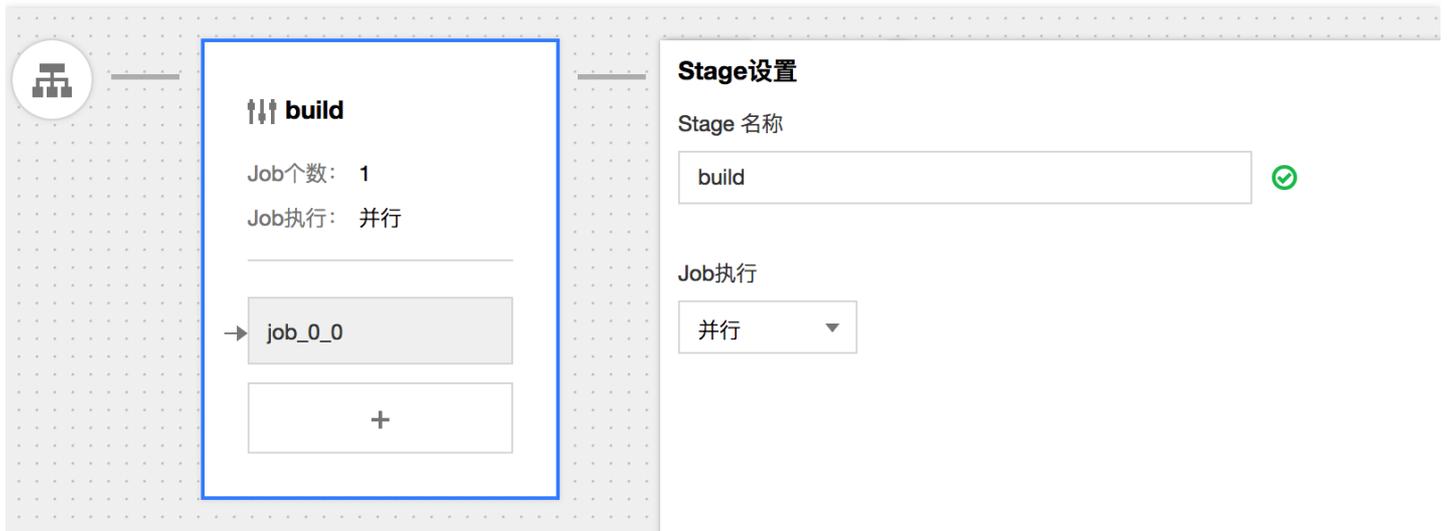
手动触发, 指定代码分支/Tag/Commit

master

可选项, 选择 workflow 关联的代码托管仓库

选择 workflow 关联代码源自动触发方式

2.新增一个 Stage , 设置 Stage 信息。



Stage设置

Stage 名称
 ✔

Job 执行

3.新增一个 Job , 设置 Job 信息, 选择 workflow 组件。



Job基础设置

Job 名称
 ✔

工作方式

workflow 组件地址 (必填)
 ↓ 选择

workflow 组件版本

允许失败

失败重试次数

4.Tencent Hub 预置了一些 workflow 组件, 并给这些组件都打上了关键词标签。可以通过组件标签来筛选出符合您需求的工作流组件。例如通过 Java 和 build 筛选出支持 Java 语言构建的工作流组件。

选择 workflow 组件



手动输入 workflow 组件路径



选择 workflow 组件

我的工作流组件

推荐 workflow 组件

标签: java | build x

多个关键字用竖线“|”分隔，多个

通过标签过滤选择 workflow 组件

[组件打包指引](#)

名称

标签



java_build_maven

java build maven

5.填写组件需要的参数。

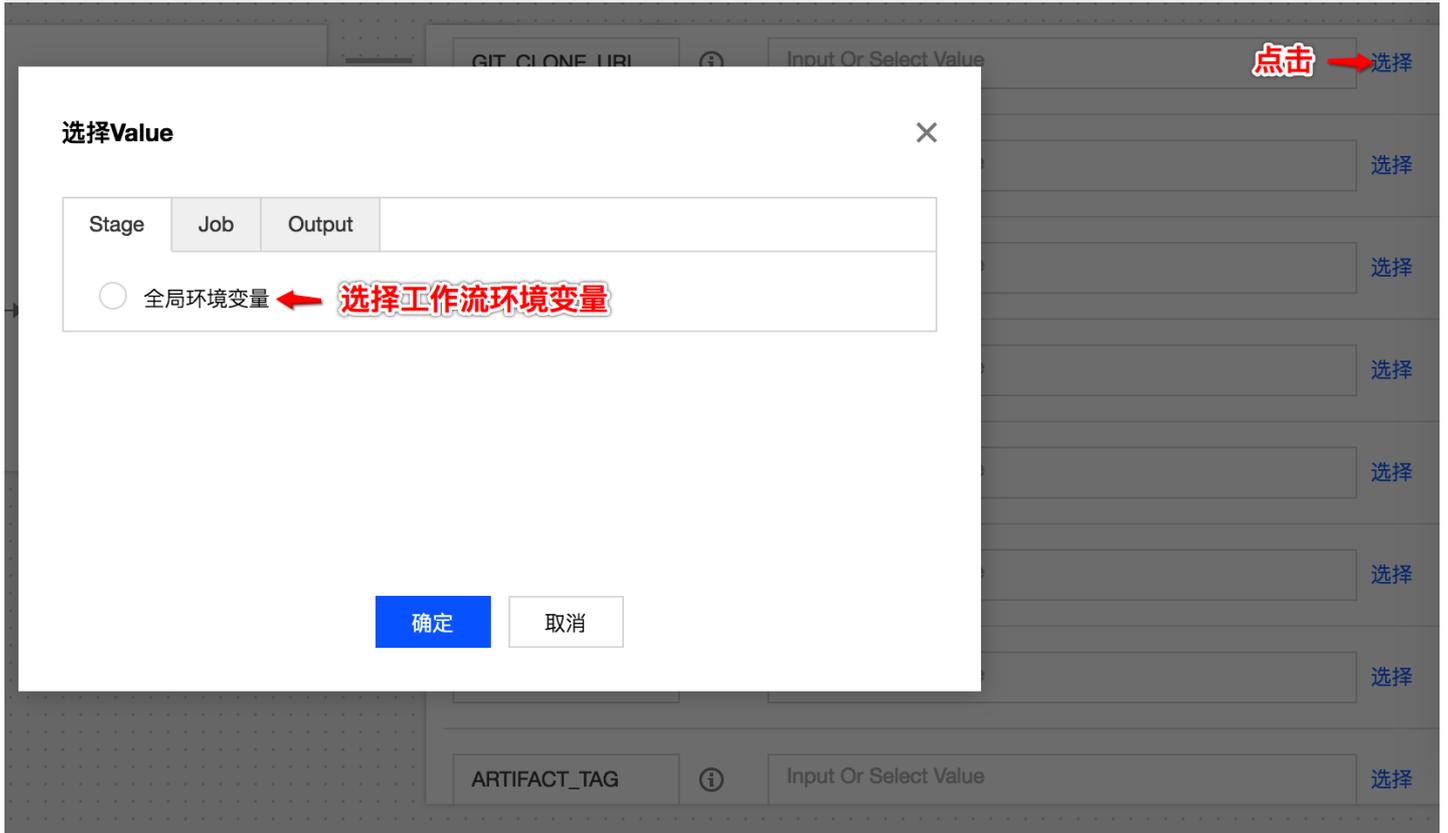
Job参数设置

INPUT参数 [填写说明](#)  **该组件功能及填写参数的详细说明**

Key		Value		
GIT_CLONE_URL		<input type="text" value="Input Or Select Value"/>	选择	删除
GIT_REF		<input type="text" value="Input Or Select Value"/>	选择	删除
GOALS		<input type="text" value="Input Or Select Value"/>	选择	删除
POM_PATH		<input type="text" value="Input Or Select Value"/>	选择	删除
HUB_REPO		<input type="text" value="Input Or Select Value"/>	选择	删除
HUB_USER		<input type="text" value="Input Or Select Value"/>	选择	删除

每个参数填写的提示信息

6.参数的值可以输入填写，也可以使用环境变量。例如，组件需要的 GIT_CLONE_URL 参数，就可以使用工作流绑定了源代码授权后生成的环境变量 `$_WORKFLOW_GIT_CLONE_URL`。



7.填写完成组件需要的参数，保存 workflow。

GIT_CLONE_URL		<code>\${_WORKFLOW_GIT_CLONE_URL}</code>
GIT_REF		<code>\${_WORKFLOW_GIT_REF}</code>
GOALS		Input Or Select Value
POM_PATH		Input Or Select Value
HUB_REPO		3321337994/maven-demo
HUB_USER		Input Or Select Value
HUB_TOKEN		Input Or Select Value
_WORKFLOW_FLAC		true
ARTIFACT_TAG		v1.0

8.运行 workflow。

项目仓库 (maven-demo)

项目仓库信息 workflow 镜像管理 文件管理 Webhook通知

[新建](#)

workflow名	最后执行状态	最后执行耗时	最后结束时间	总执行数(次)	生效 ⓘ	操作
▶ workflow_java_bu...	-	-	-	0	<input type="checkbox"/>	立即执行

9.运行成功后，可以看到项目仓库的文件管理下，自动生成了构建结果。

workflow名	最后执行状态	最后执行耗时	最后结束时间	总执行数(次)	生效 ⓘ
▶ workflow_java_bu...	成功	1分钟59秒	2018-08-01 16:1...	2	<input checked="" type="checkbox"/>

[←](#) **项目仓库 (maven-demo)**

项目仓库信息 workflow 镜像管理 文件管理 Webhook通知

~/

名称	版本	更新时间
▶ maven_demo.tar.bz	v1.0	2018-08-01 17:39:19

容器服务 CICD workflow

最近更新时间：2019-09-11 11:48:19

实践-容器服务CI/CD workflow

本文主要叙述如何通过 Tencent Hub 实现容器服务 CI/CD workflow，以部署开源 PHP 游戏网页游戏 2Moons 为例。

前期资源

- [游戏源码](#)
- [Tencent Hub](#)
- [TKE](#)
- 预先准备一个备案过的域名(eg:game.tdemo.net 或直接使用 IP)

说明：

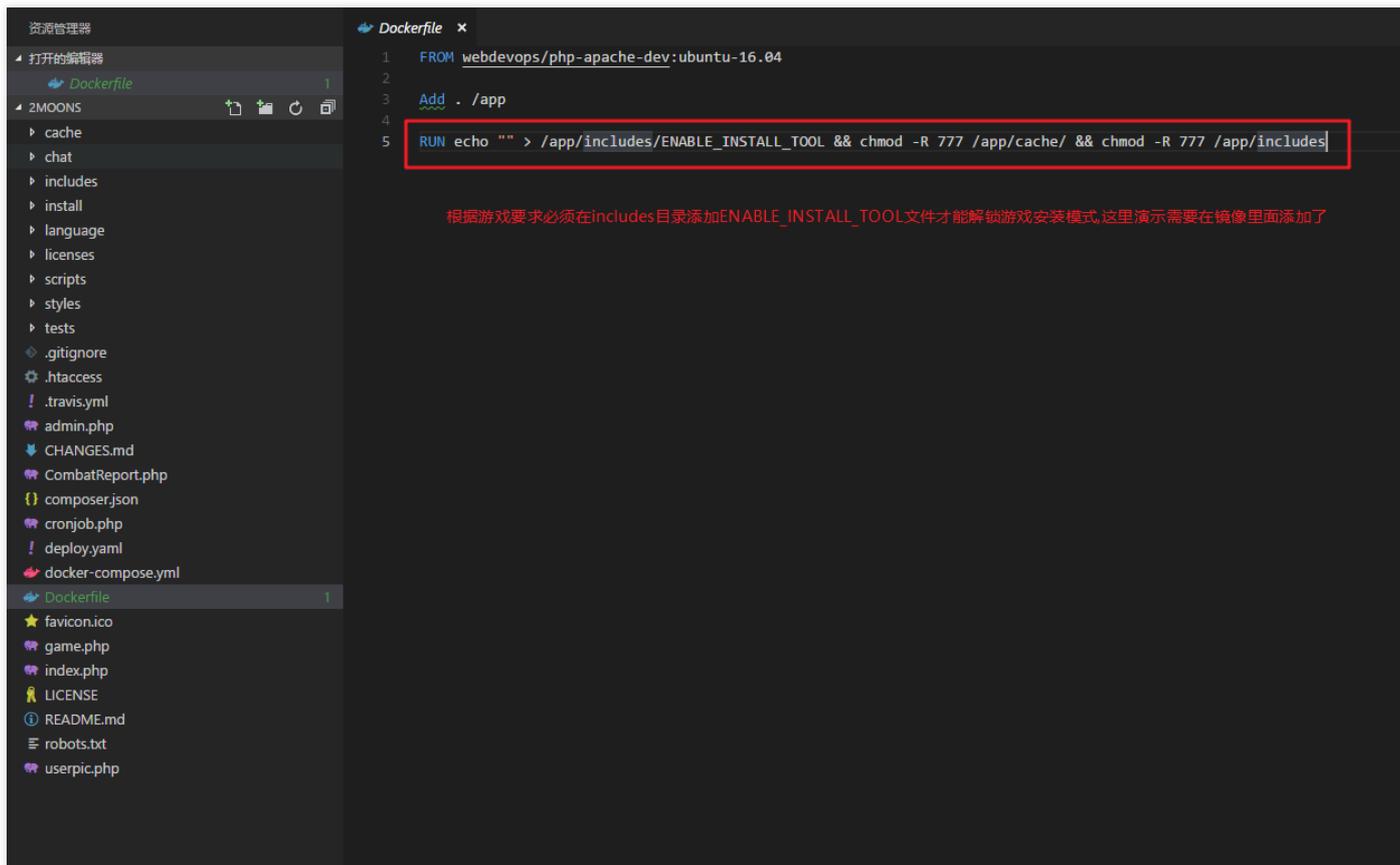
例子中为了方便测试，数据库也部署在 TKE 里面，生产环境一般会使用 TencentDB 实例。

准备工作

1. fork 游戏源码到自己的 github 方便后续修改和提交。这里 fork 到 <https://github.com/Kevin-Qiu/2Moons> 并新建 `tencent_cloud` 分支。
2. 准备好 Dockerfile，可以参考 https://github.com/Kevin-Qiu/2Moons/blob/tencent_cloud/docker-compose.yml。

```
1 version: '2'
2 services:
3   web:
4     image: webdevops/php-apache-dev:ubuntu-16.04
5     depends_on:
6       - db
7     volumes:
8       - ./app
9     environment:
10      DB_HOST: database
11      DB_USER: 2moons
12      DB_PASSWORD: 2moons
13      DB_NAME: 2moons
14      ADMIN_NAME: admin
15      ADMIN_PASSWORD: admin
16      ADMIN_MAIL: mail@example.com
17     ports:
18       - 8080:80
19     networks:
20       - 2moons
21   db:
22     image: mariadb
23     environment:
24       MYSQL_ROOT_PASSWORD: root
25       MYSQL_DATABASE: 2moons
26       MYSQL_USER: 2moons
27       MYSQL_PASSWORD: 2moons
28     ports:
29       - 3306:3306
30     networks:
31       2moons:
32         aliases:
33           - database
34
35   networks:
36     2moons:
```

https://github.com/Kevin-Qiu/2Moons/blob/tencent_cloud/Dockerfile



```
1 FROM webdevops/php-apache-dev:ubuntu-16.04
2
3 Add . /app
4
5 RUN echo "" > /app/includes/ENABLE_INSTALL_TOOL && chmod -R 777 /app/cache/ && chmod -R 777 /app/includes
```

根据游戏要求必须在includes目录添加ENABLE_INSTALL_TOOL文件才能解锁游戏安装模式,这里演示需要在镜像里面添加了

3. 准备好 TKE 集群，对于这个例子来说，集群里面建议最好有4核 8G 的空闲资源，集群开启外网访问并记下账号/密码/证书/地址。

总览
云产品
云服务器
域名注册
网站备案
容器服务
Tencent Hub

容器服务

应用中心

- ◆ 集群
- 黑石集群
- 应用
- 服务
- Ingress

交付中心

- 镜像仓库
- 应用模板仓库

配置项
运维中心

- 日志

体验&指引

- 免费实验室
- 使用指引

(temp)

节点列表 Namespace列表 伸缩组列表 集群信息

基本信息

集群名称	temp
新增资源所属项目	默认项目
集群ID	
状态	运行中
k8s版本	1.10.5
节点数量	2个
配置	7.84核 28.36GB
所在地域	华南地区(广州)
节点网络	
容器网络	172.19.0.0/16
集群凭证	显示凭证
创建时间	2018-07-18 22:18:54
更新时间	2018-07-18 23:22:25
描述	无

高级设置

ipvs支持 未开启

集群凭证 ✕

安装Kubectl后，您可以通过用户名密码或集群CA证书登录到集群 [Kubectl帮助](#)

用户名

密码

集群CA证书 下载 复制

```
-----BEGIN CERTIFICATE-----  
[REDACTED]  
-----END CERTIFICATE-----
```

外网访问地址 已开启

内网访问地址 未开启

4. 预先在 TKE 里面部署好 2Moon 需要的数据库，服务名 mariadb，1C2G，环境变量可以参考 https://github.com/Kevin-Qiu/2Moons/blob/tencent_cloud/docker-compose.yml，端口 3306，只允许集群内访问。

Branch: master 2Moons / docker-compose.yml

jkroepke Fixed install issues

1 contributor

36 lines (35 sloc) | 660 Bytes

```
1  version: '2'
2  services:
3    web:
4      image: webdevops/php-apache-dev:ubuntu-16.04
5      depends_on:
6        - db
7      volumes:
8        - ./app
9      environment:
10     DB_HOST: database
11     DB_USER: 2moons
12     DB_PASSWORD: 2moons
13     DB_NAME: 2moons
14     ADMIN_NAME: admin
15     ADMIN_PASSWORD: admin
16     ADMIN_MAIL: mail@example.com
17     ports:
18       - 8080:80
19     networks:
20       - 2moons
21   db:
22     image: mariadb
23     environment:
24       MYSQL_ROOT_PASSWORD: root
25       MYSQL_DATABASE: 2moons
26       MYSQL_USER: 2moons
27       MYSQL_PASSWORD: 2moons
28     ports:
29       - 3306:3306
30     networks:
31       2moons:
32         aliases:
33           - database
34
35   networks:
36     2moons:
```

← 新建服务

基本信息

服务名称

最长40个字符，只能包含小写字母、数字及分隔符("-")，且必须以小写字母开头，数字或小写字母结尾

所在地域

运行集群

(temp)

如现有的集群不合适，您可以去控制台 [新建集群](#)

服务描述

请输入描述信息，不超过1000个字符

部署设置(Deployment)

数据卷(选填) ⓘ

[添加数据卷](#)

为容器提供存储，目前支持临时路径、主机路径、云硬盘数据卷、文件存储NFS、配置项，还需挂载到容器的指定路径中。 [使用指引](#)

运行容器

✓ ✕

名称 最长63个字符，只能包含小写字母、数字及分隔符("-")，且不能以分隔符开头或结尾

镜像 [选择镜像](#)

镜像版本 (Tag)

CPU/内存限制

CPU限制				内存限制							
request	1	-	limit	1	核	request	2048	-	limit	2048	MB

Request用于预分配资源,当集群中的节点没有request所要求的资源数量时,容器会创建失败。
Limit用于设置容器使用资源的最大上限,避免异常情况下节点资源消耗过多。

环境变量 ⓘ

MYSQL_ROOT_PA	=	root	✕
MYSQL_DATABAS	=	2moons	✕
MYSQL_USER	=	2moons	✕
MYSQL_PASSWOF	=	2moons	✕

[新增变量](#) [从配置项导入](#)

变量名只能包含大小写字母、数字及下划线，并且不能以数字开头

[显示高级设置](#)

注意：服务创建完成后，容器的配置信息可以通过更新服务的方式进行修改

访问设置(Service)

服务访问方式 ^①

- 提供公网访问
 仅在集群内访问
 VPC内网访问
 不启用（不支持Ingress）
 [如何选择](#)

将提供一个可以被集群内其他服务或容器访问的入口，支持TCP/UDP协议，数据库类服务如Mysql可以选择集群内访问,来保证服务网络隔离性支持Ingress。 [查看详情](#)

端口映射

协议 ^①	容器端口	服务端口 ^①
TCP	3306	3306

[添加端口映射](#)

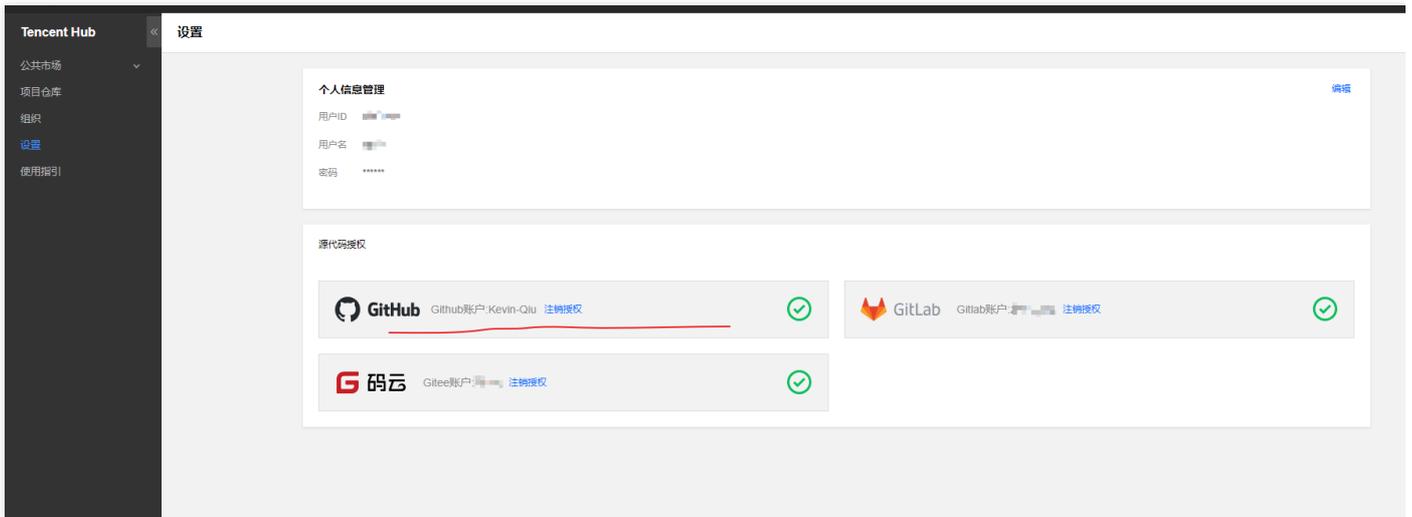
- 概览
- 应用中心
- 集群
- 黑石集群
- 应用
- 服务**
- Ingress
- 交付中心
- 镜像仓库
- 应用模板仓库
- 配置项

新建

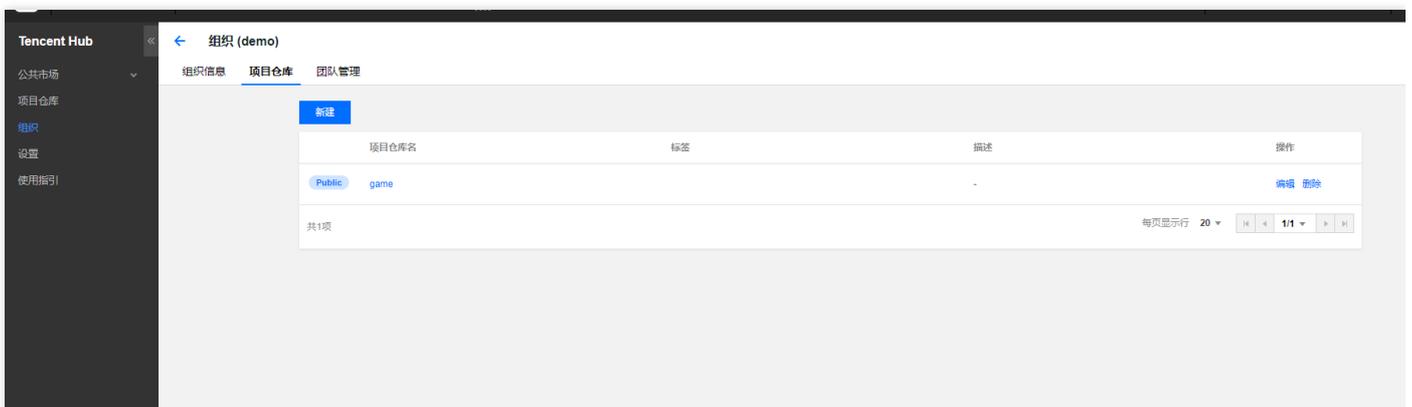
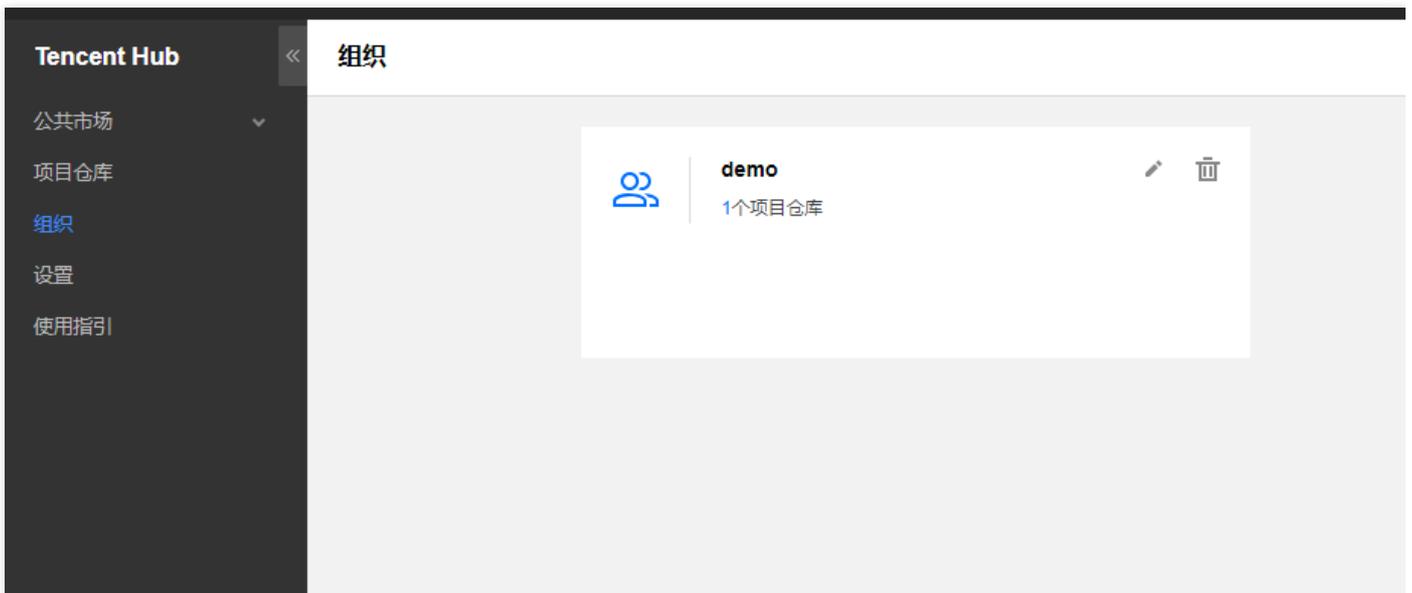
名称 ^①	监控	日志	状态	运行/预期数量	IP地址 ^①	负载均衡	标签(label) [⌵]
mariadb			运行中	1/1个	172.19.255.10	未启用	qcloud-app:mariadb,

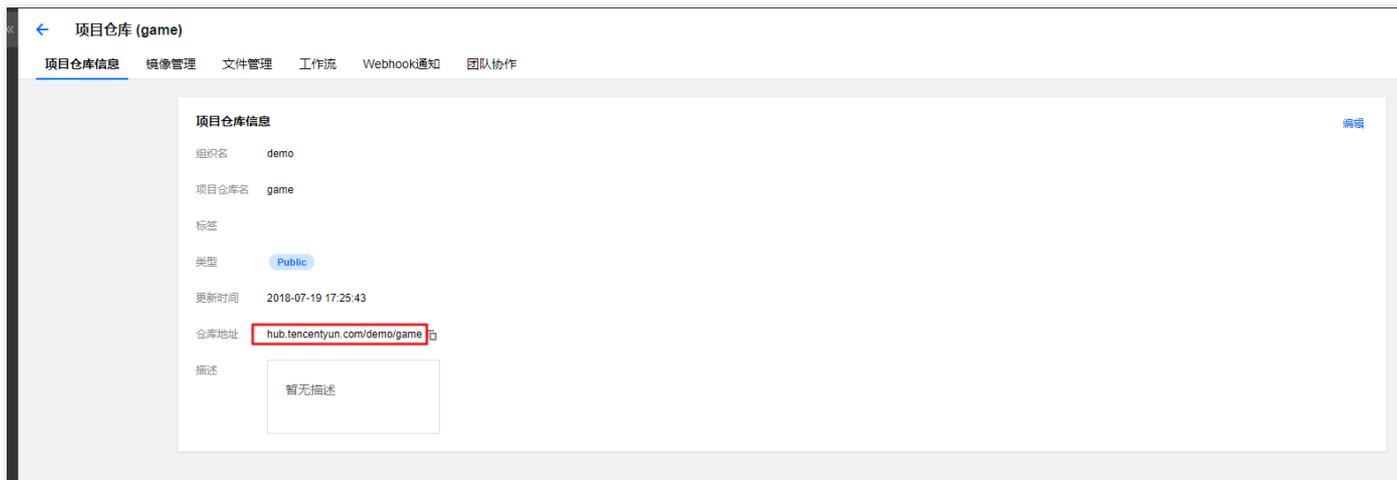
持续集成 CI

1. 先在 Tencent Hub 里面绑定 github。



2. 在 Tencent Hub 里面建立一个组织和仓库 demo/game，并记录下仓库地址/镜像地址。





3. 准备好需要部署到 TKE 的 yaml 文件。

https://github.com/Kevin-Qiu/2Moons/blob/tencent_cloud/deploy.yaml

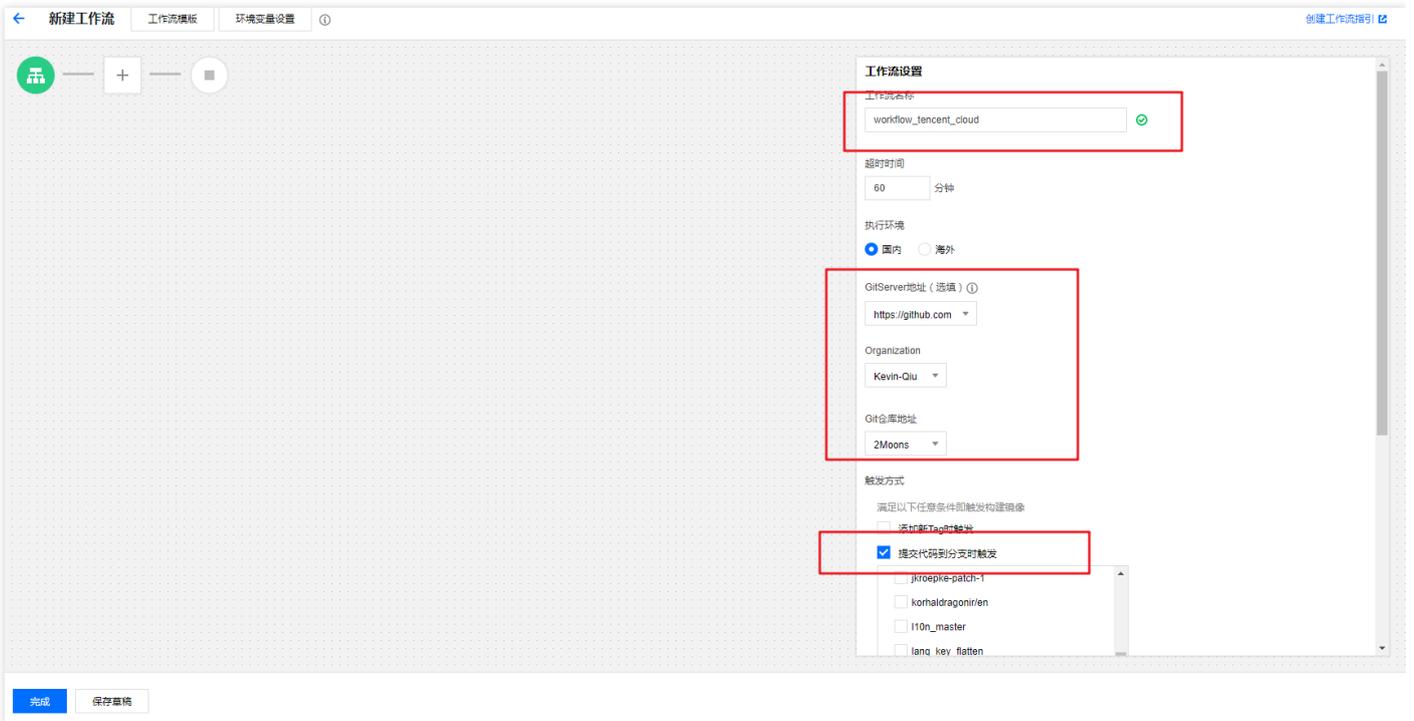
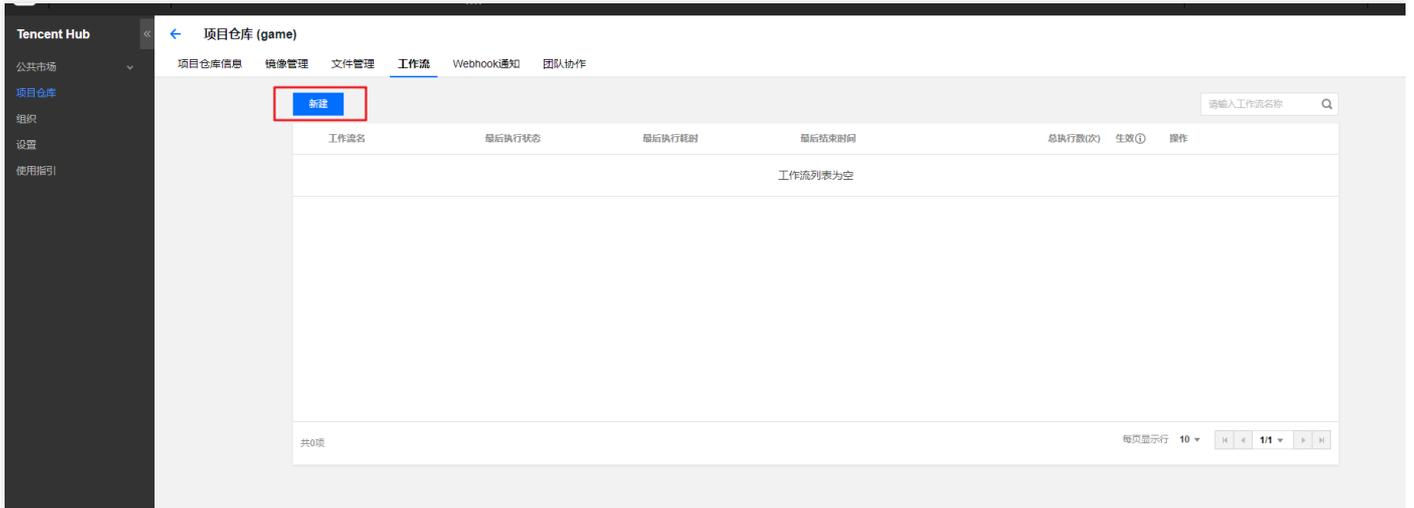
```

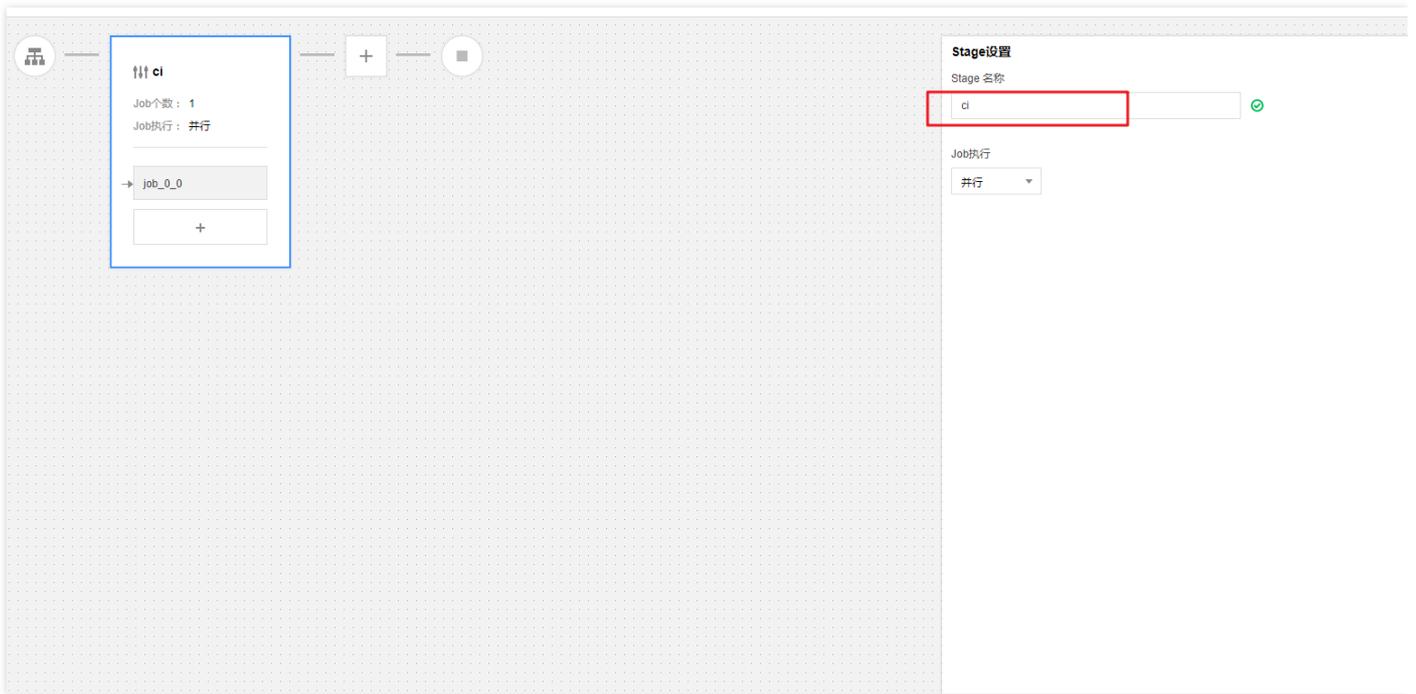
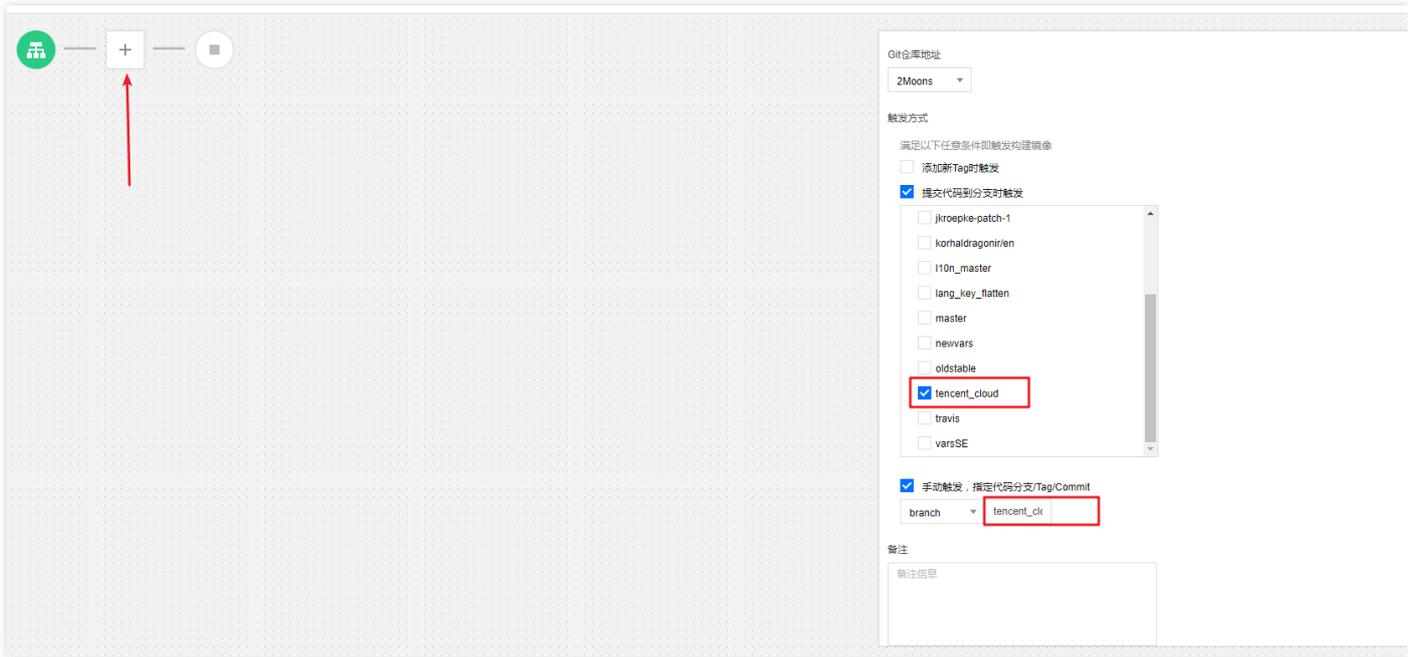
! deploy.yaml x
1  apiVersion: extensions/v1beta1
2  kind: Deployment
3  metadata:
4    labels:
5      qcloud-app: moon
6    name: moon
7    namespace: default
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       qcloud-app: moon
13   template:
14     metadata:
15       labels:
16         qcloud-app: moon
17     spec:
18       containers:
19         - env:
20           - name: DB_HOST
21             value: mariadb
22           - name: DB_USER
23             value: 2moons
24           - name: DB_PASSWORD
25             value: 2moons
26           - name: DB_NAME
27             value: 2moons
28           - name: ADMIN_NAME
29             value: admin
30           - name: ADMIN_PASSWORD
31             value: admin
32           - name: ADMIN_MAIL
33             value: admin@tdemo.net
34   image: hub.tencentyun.com/demo/game@$IMAGE_DIGEST
35   imagePullPolicy: Always
36   name: moon
37   resources:
38     limits:
39       cpu: "2"
40       memory: 2Gi
41     requests:
42       cpu: "1"
43       memory: 1Gi
44   imagePullSecrets:
45     - name: qcloudregistrykey
46     - name: tencenthubkey
47

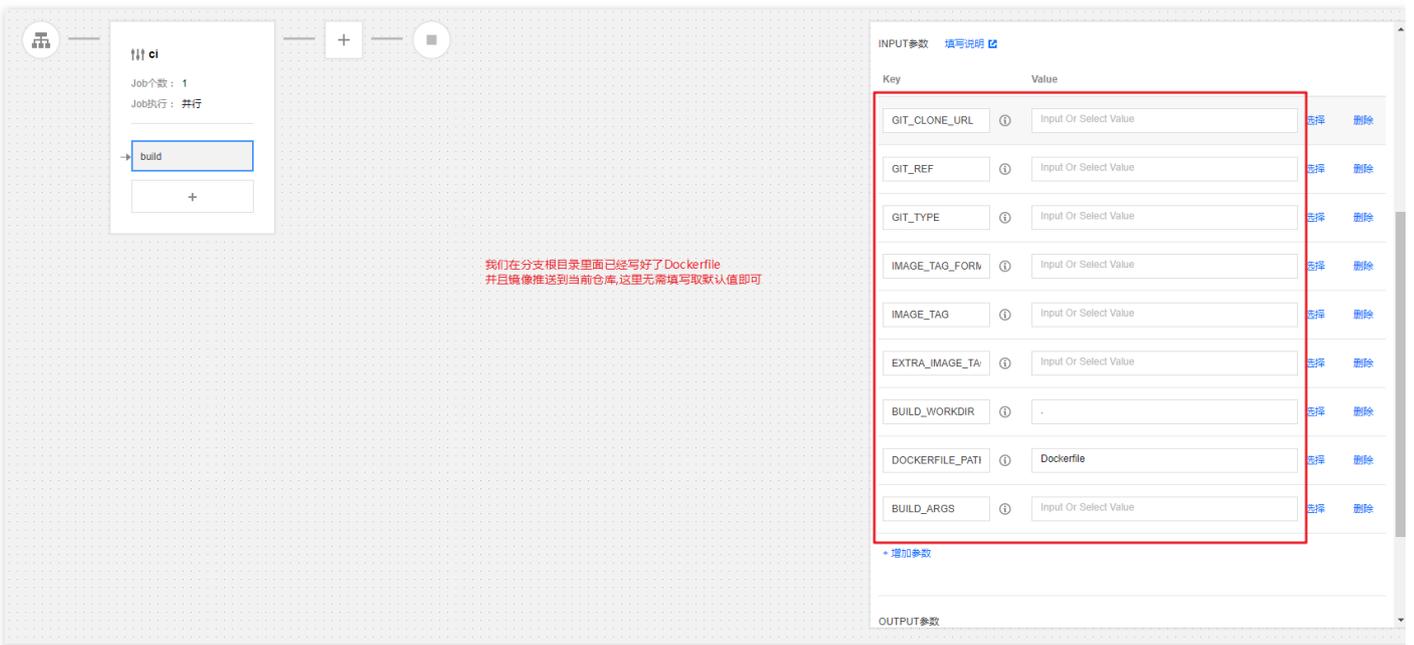
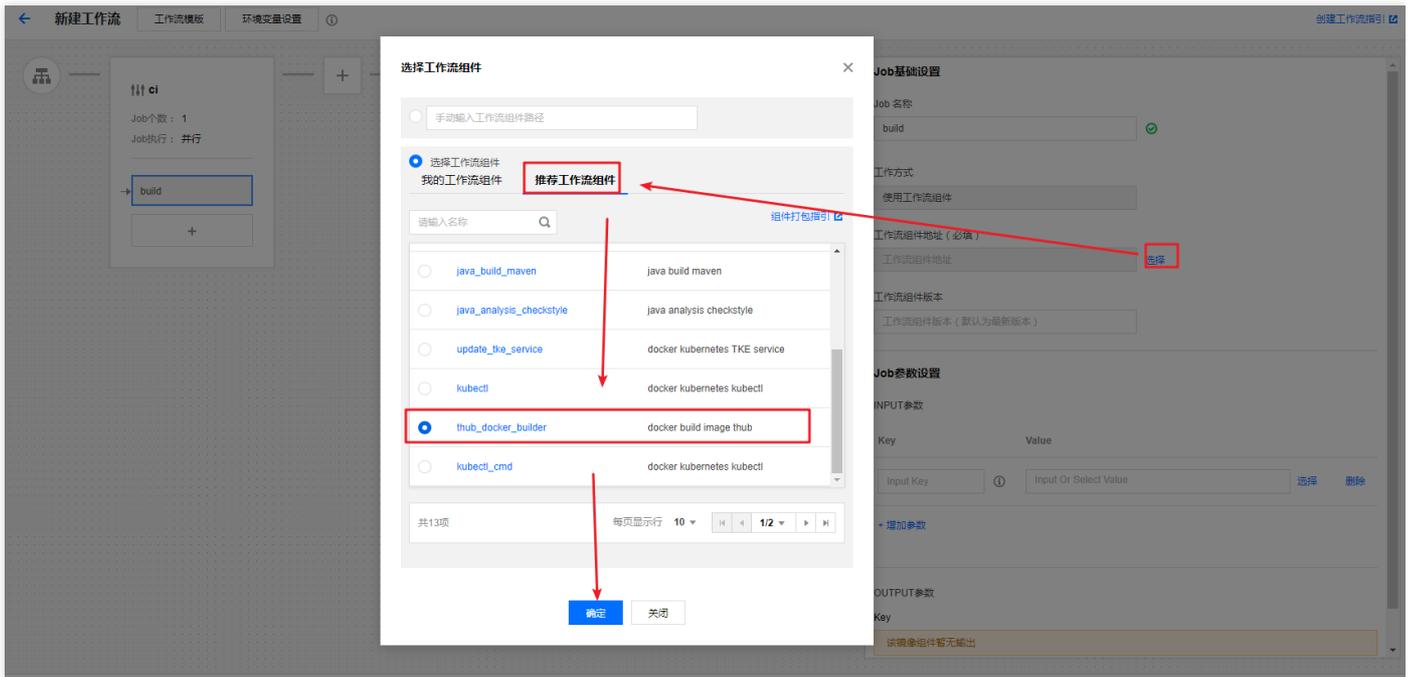
```

```
43     memory: 1Gi
44     imagePullSecrets:
45     - name: qcloudregistrykey
46     - name: tencenthubkey
47 ---
48 apiVersion: v1
49 kind: Service
50 metadata:
51   labels:
52     qcloud-app: moon
53   name: moon
54   namespace: default
55 spec:
56   type: NodePort
57   ports:
58   - name: tcp-80-80
59     port: 80
60     protocol: TCP
61     targetPort: 80
62   selector:
63     qcloud-app: moon
64 ---
65 apiVersion: extensions/v1beta1
66 kind: Ingress
67 metadata:
68   name: game
69   namespace: default
70 spec:
71   rules:
72   - host: game.tdemo.net
73     http:
74       paths:
75       - backend:
76           serviceName: moon
77           servicePort: 80
78     path: /
```

4. 在仓库下面建立 workflow，填入 workflow 名称，选择 github 项目，绑定自动和手动触发分支。







5.推送代码到 <https://github.com/Kevin-Qiu/2Moons> 的 `Tencent_Cloud` 分支测试一下前面的 CI 结果，单击查看执行历史可以查看执行明细。

新建 请输入工作流名称 🔍

工作流名	最后执行状态	最后执行耗时	最后结束时间	总执行次数	生效	操作
workflow_tencent_cloud	成功	46秒	2018-08-02 15:36:06	1	<input checked="" type="checkbox"/>	立即执行 查看与编辑 更多

工作流执行历史

ID	状态	最后执行耗时	最后结束时间	操作
2082	成功	46秒	2018-08-02 15:36:06	查看执行历史

共1项 每页显示行 10

共1项 每页显示行 10

workflow_tencent_cloud执行结果查询

Job个数: 1
Job执行: 并行

build

执行成功
 执行失败
 执行中
 执行耗时

build 导出日志

执行日志 执行参数

```

1 Step 3/3 : RUN echo "" > /app/includes/ENABLE_INSTALL_TOOL.sh chnod -> ??? /app/cache/ $0 chnod
  -> ??? /app/includes
2 --> Running in 2f3272e04fae
3 Removing intermediate container 2f3272e04fae
4 --> f881e0e8b648
5 Successfully built f881e0e8b648
6 Successfully tagged hub.tencentyun.com/demo/gane:latest
7
8 Run docker build succeeded
9 [2018-08-02 07:36:40] Run CMD: docker push hub.tencentyun.com/demo/gane:latest
10 The push refers to repository [hub.tencentyun.com/demo/gane]
11 8c5ee41c6182: Preparing
12 3a6b84988001: Preparing
13 6781eef78b52: Preparing
14 788e7843742d: Preparing
15 e01846847a83: Preparing
16 20303885535a: Preparing
17 542e437351fa: Preparing
18 234b4d811b57: Preparing
19 6f928e276267: Preparing
20 e93e4d1d6e79: Preparing
21 62e511e85e0: Preparing
22 949e3ee3fe2c: Preparing
23 5ee3e877fa: Preparing
24 f88e27f90d9e: Preparing
25 7881d3001e4: Preparing
26 078e43e04ebd: Preparing
27 6ee3d4f893f1: Preparing
28 e4e18820e278: Preparing
29 711e4cb2f50: Preparing
30 20303885535a: Waiting
31 f88e27f90d9e: Waiting
32 542e437351fa: Waiting
33 7881d3001e4: Waiting
34 234b4d811b57: Waiting
35 078e43e04ebd: Waiting
36 6f928e276267: Waiting
    
```

workflow_tencent_cloud执行结果查询

ci

Job个数: 1

Job执行: 并行

build

build

执行日志 执行参数

输入参数

Key	Value
BUILD_ARGS	无
BUILD_WORKDIR	
DOCKERFILE_PATH	Dockerfile

输出参数

Key	Value
IMAGE	hub.tencentyun.com/demo/game
IMAGE_DIGEST	sha256:9a1d7c4ceaaa8dc1f9271717580054683fa62...
IMAGE_ID	sha256:881c0e8d6487e8b89d0f36ac4eaa82244af3...

■ 执行成功

■ 执行失败

或者可以手动单击立即执行进行手动构建。

项目仓库 (game)

项目仓库信息 workflows 镜像管理 文件管理 Webhook通知 团队协作

请输入工作流名称

工作流名	最后执行状态	最后执行耗时	最后结束时间	总执行数(次)	生效	操作
workflow_tencent_cloud	等待执行	-	-	2	<input checked="" type="checkbox"/>	立即执行

工作流执行历史

ID	状态	最后执行耗时	最后结束时间	操作
2104	等待执行	-	-	查看执行历史 终止
2082	成功	46秒	2018-08-02 15:36:06	查看执行历史

共2项 每页显示 10

共1项 每页显示 10

新建 请输入 workflows 名称

工作流名称	最后执行状态	最后执行耗时	最后结束时间	总执行次数(次)	生效①	操作
workflow_tencent_cloud	成功	44秒	2018-08-02 20:22:03	2	<input checked="" type="checkbox"/>	立即执行 查看与编辑 更多

工作流执行历史

ID	状态	最后执行耗时	最后结束时间	操作
2104	成功	44秒	2018-08-02 20:22:03	查看执行历史
2082	成功	46秒	2018-08-02 15:36:06	查看执行历史

共2项 每页显示 10

共1项 每页显示 10

← 项目仓库 (game)

项目仓库信息 工作流 **镜像管理** 文件管理 Webhook通知 团队协作

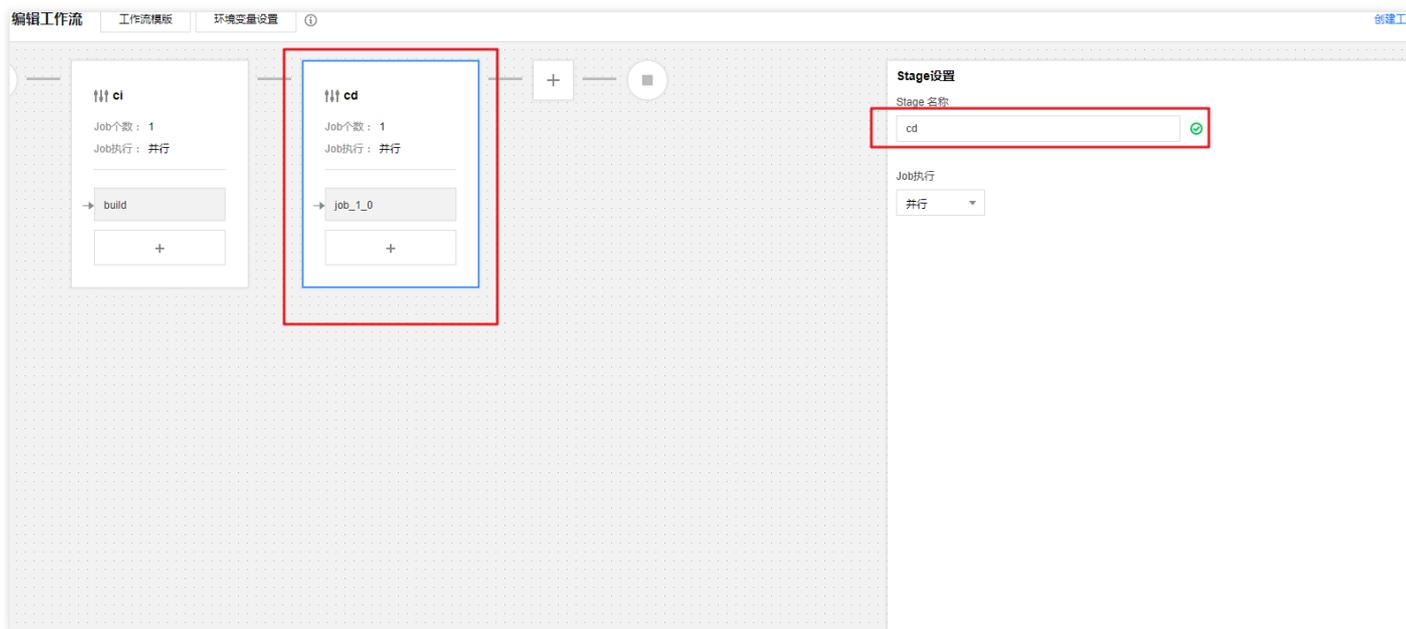
使用指引 | 镜像构建 | 触发器

ID	标签	更新时间	大小(MB)	下载次数	操作
sha256:ba20c95d67279b580d838dd1286c074...	latest	2018-08-02 20:21:59	249	0	创建服务 删除

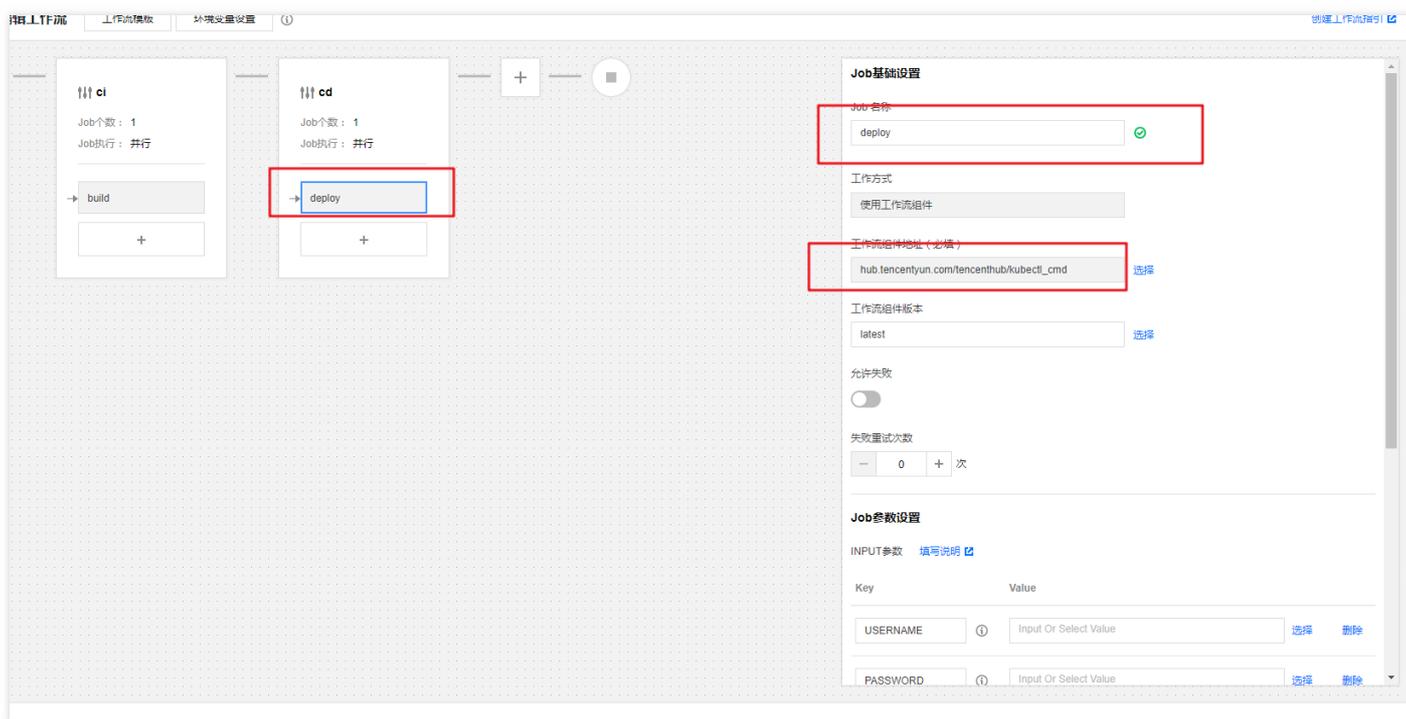
共1项 每页显示 10

持续部署 CD

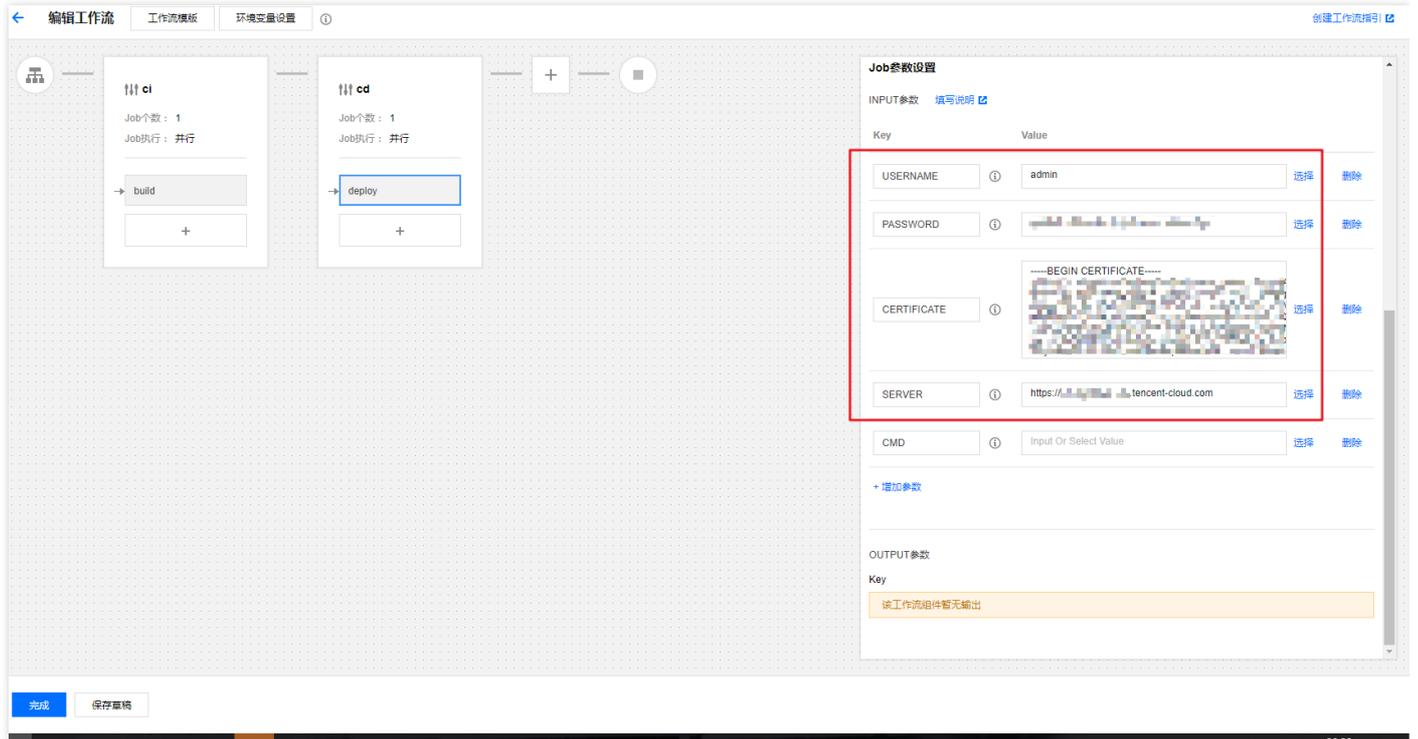
1. 单击查看与编辑继续完成 CD 部分，创建一个 CD 的 stage。



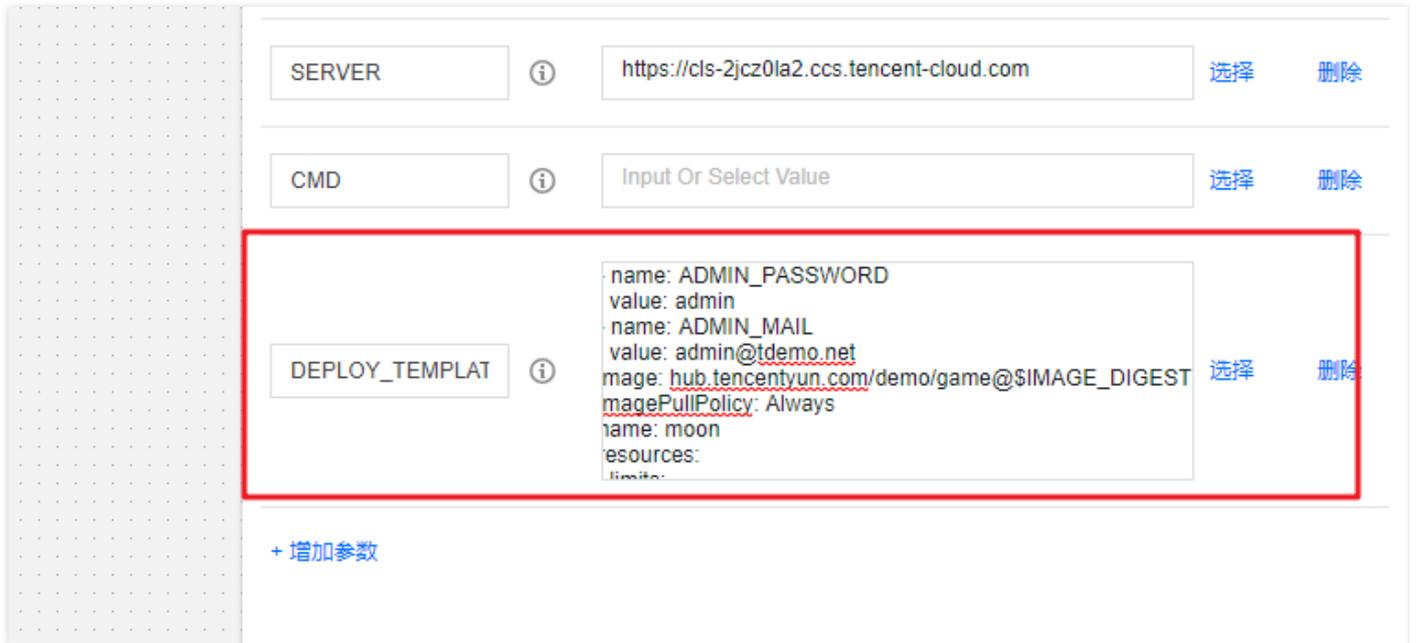
2. 修改 Job 名为 deploy，选择工作流组件为 kubectl_cmd 组件。



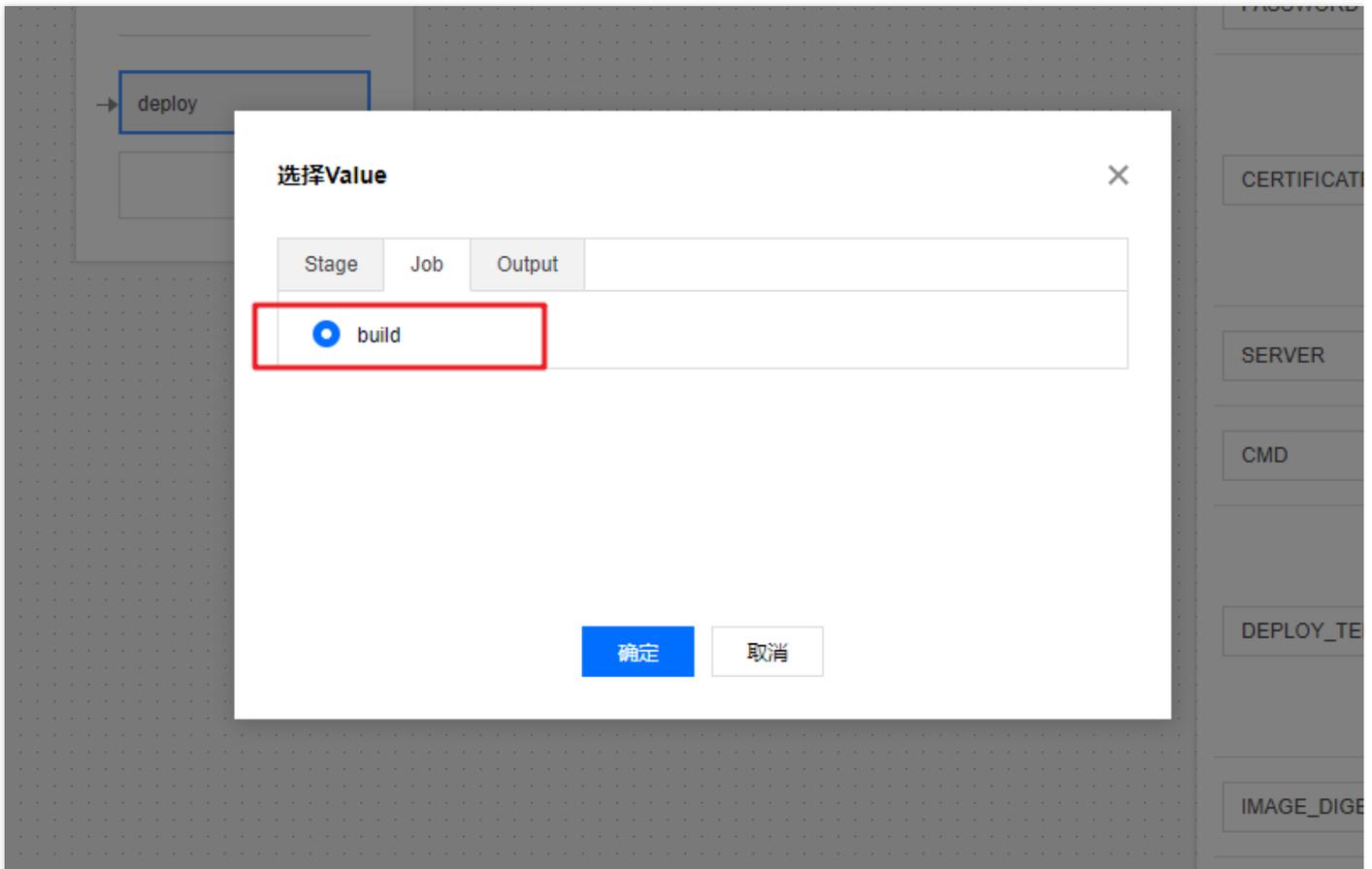
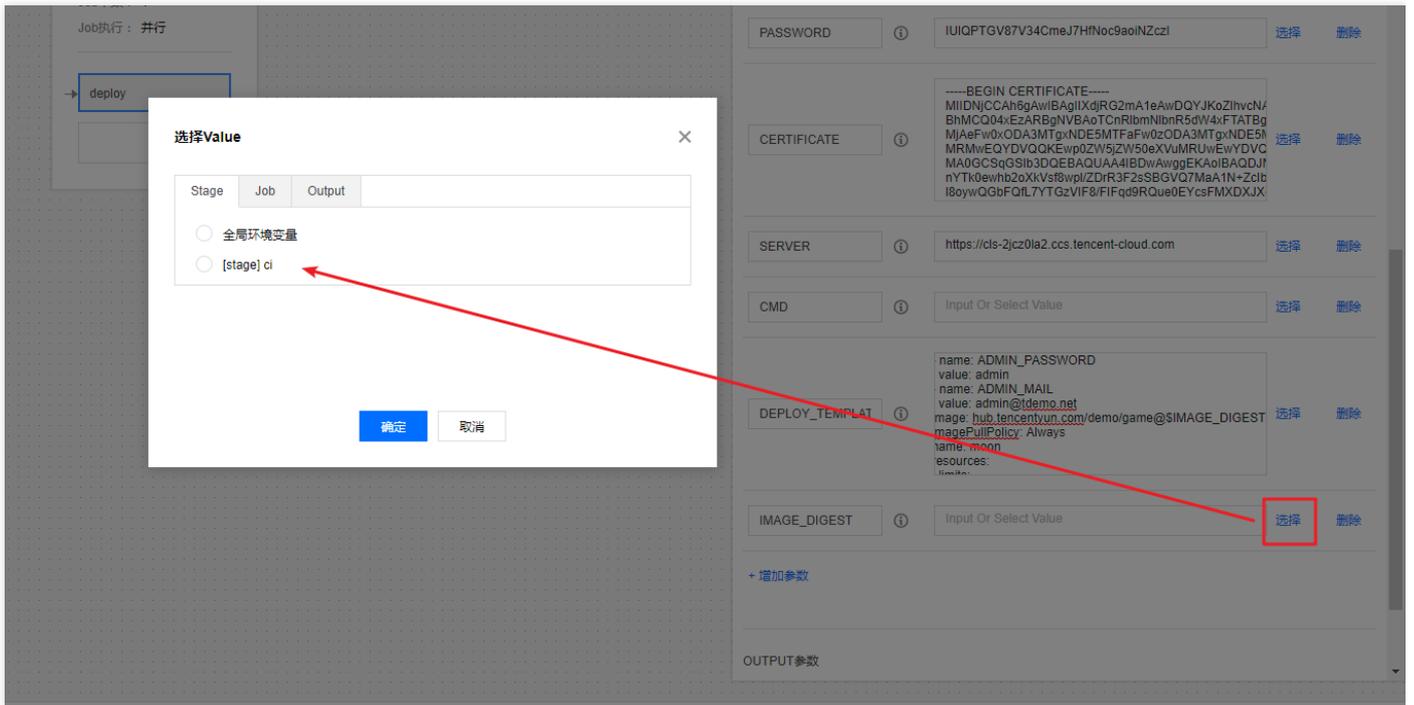
3. 把前面准备好的集群信息填入环境变量。

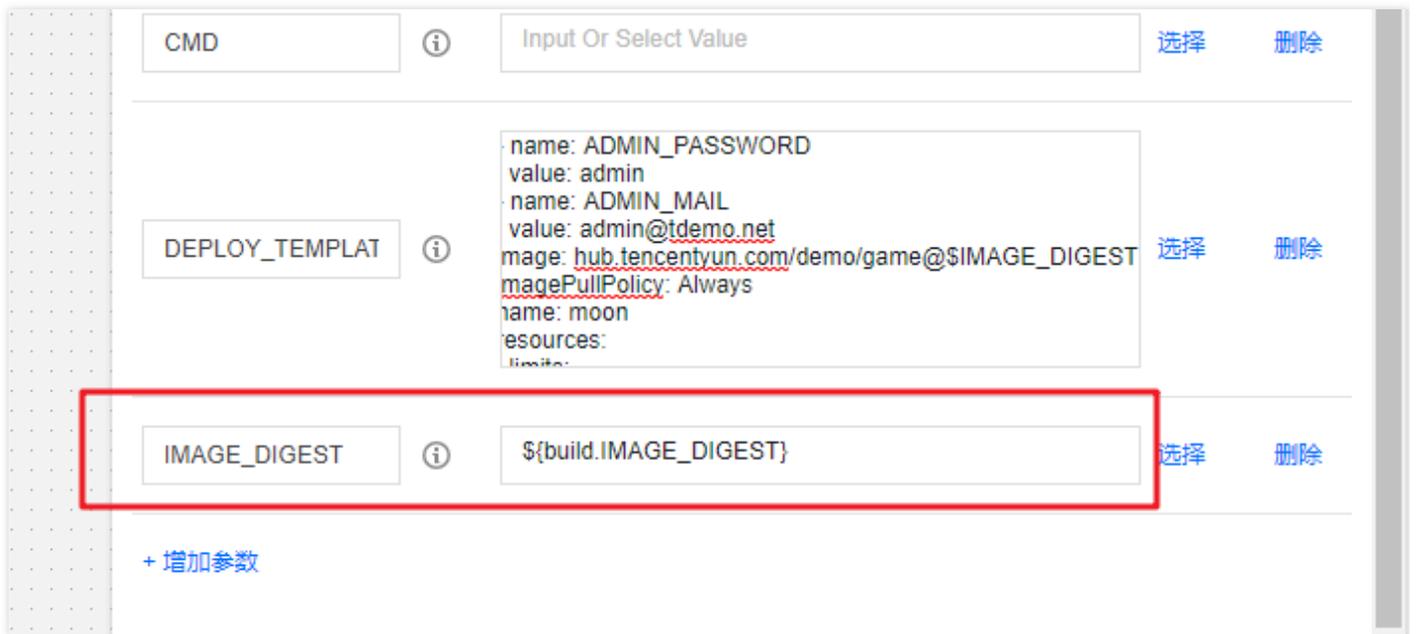
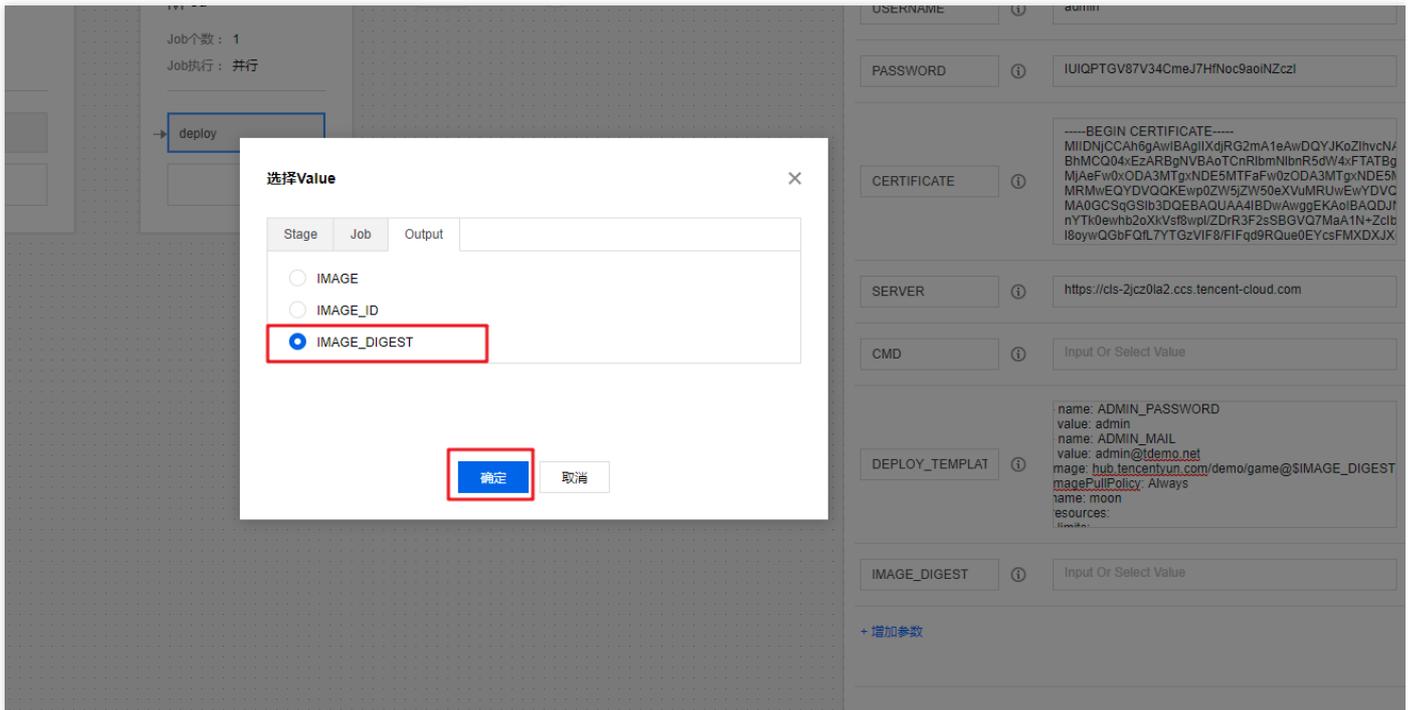


4. 把前面准备好的 deploy 模板填入输入参数(这里支持多行文本)。



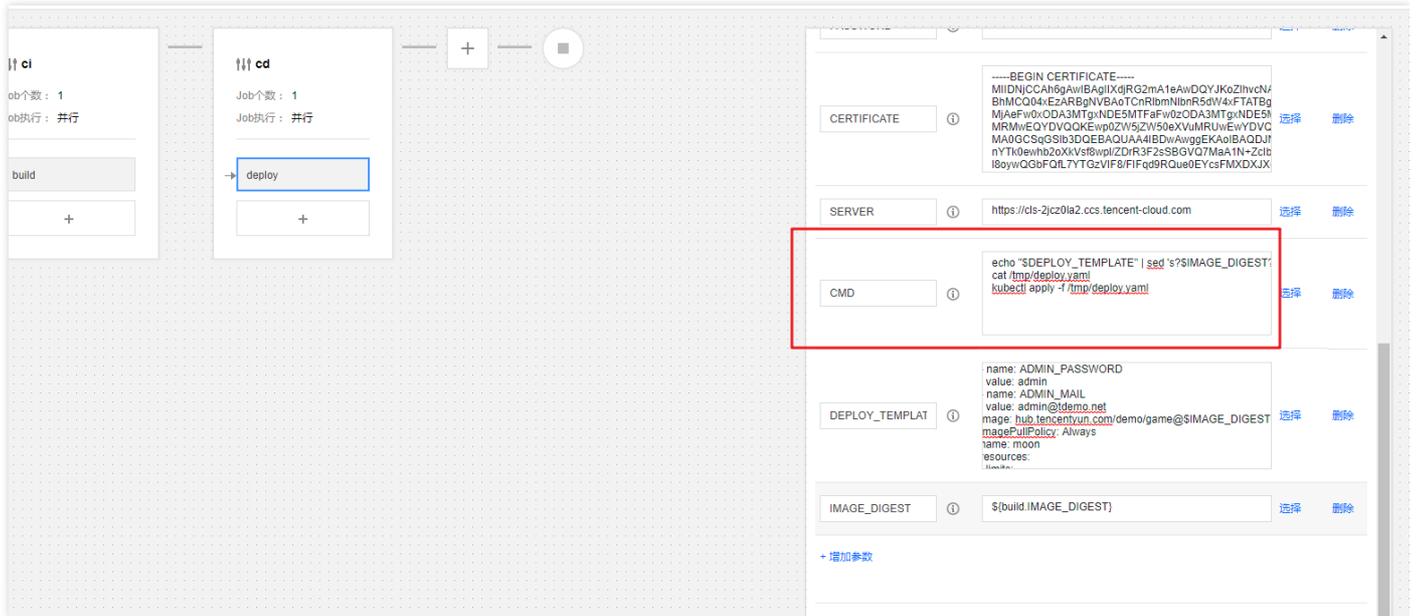
5. 从前面的 Job 里面选择输出注入到输入参数 IMAGE_DIGEST 中。





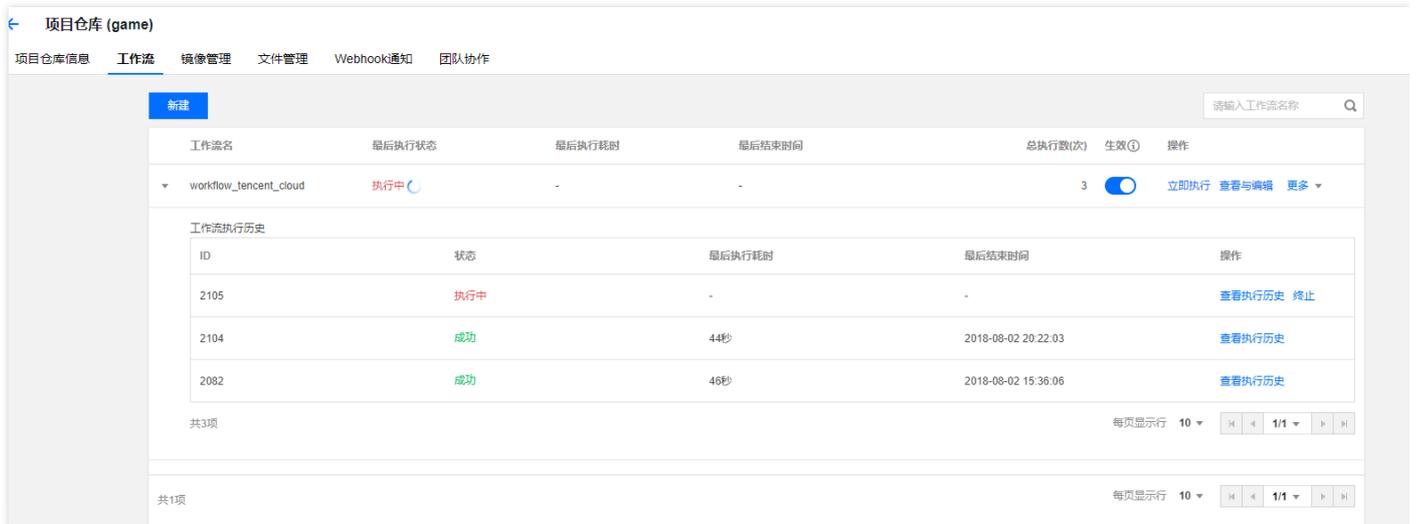
6. 最后构建 CD 的执行命令,参考命令如下 :

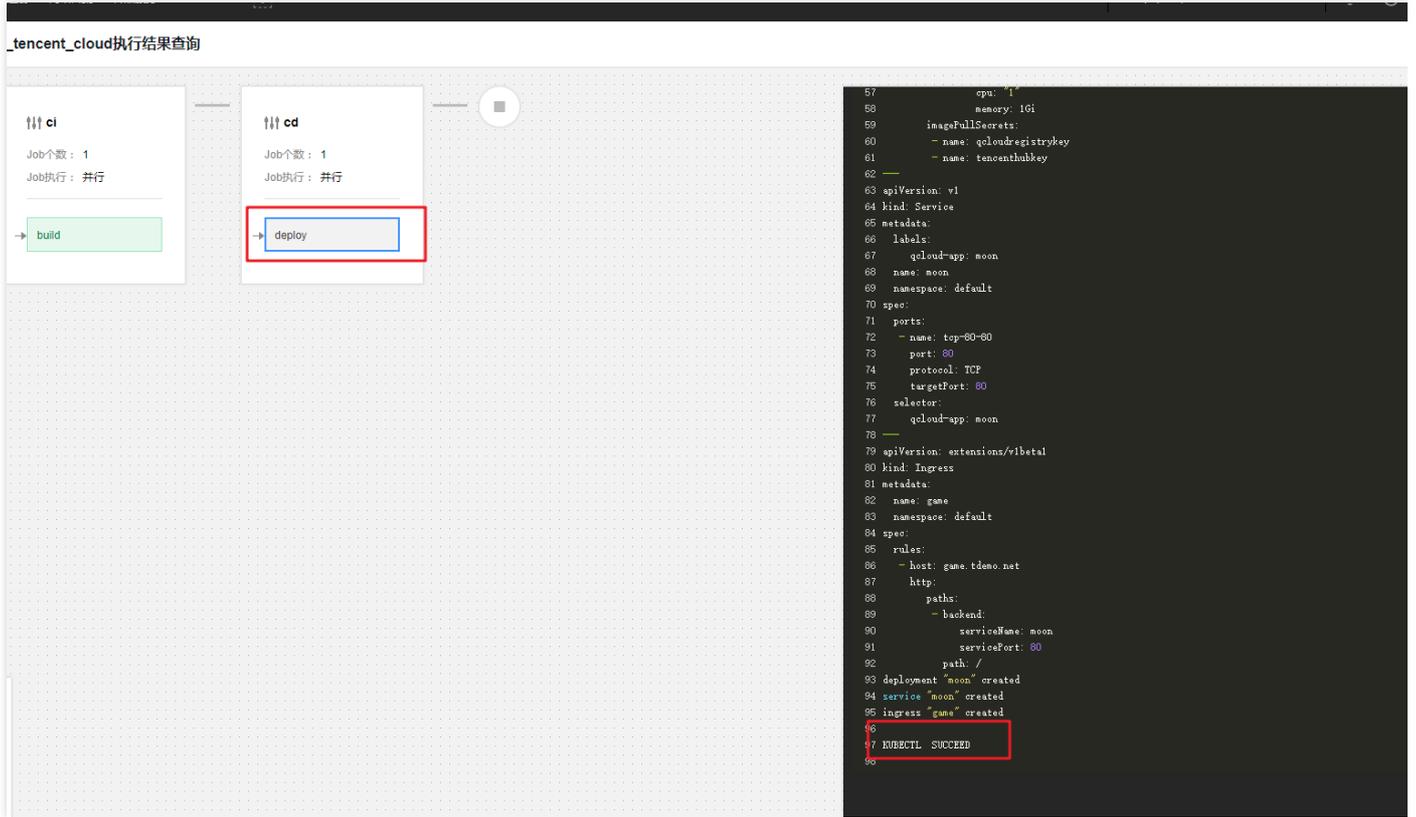
```
echo "$DEPLOY_TEMPLATE" | sed 's?${IMAGE_DIGEST}?"${IMAGE_DIGEST}"?g' >> /tmp/deploy.yaml
cat /tmp/deploy.yaml
kubectl apply -f /tmp/deploy.yaml
```



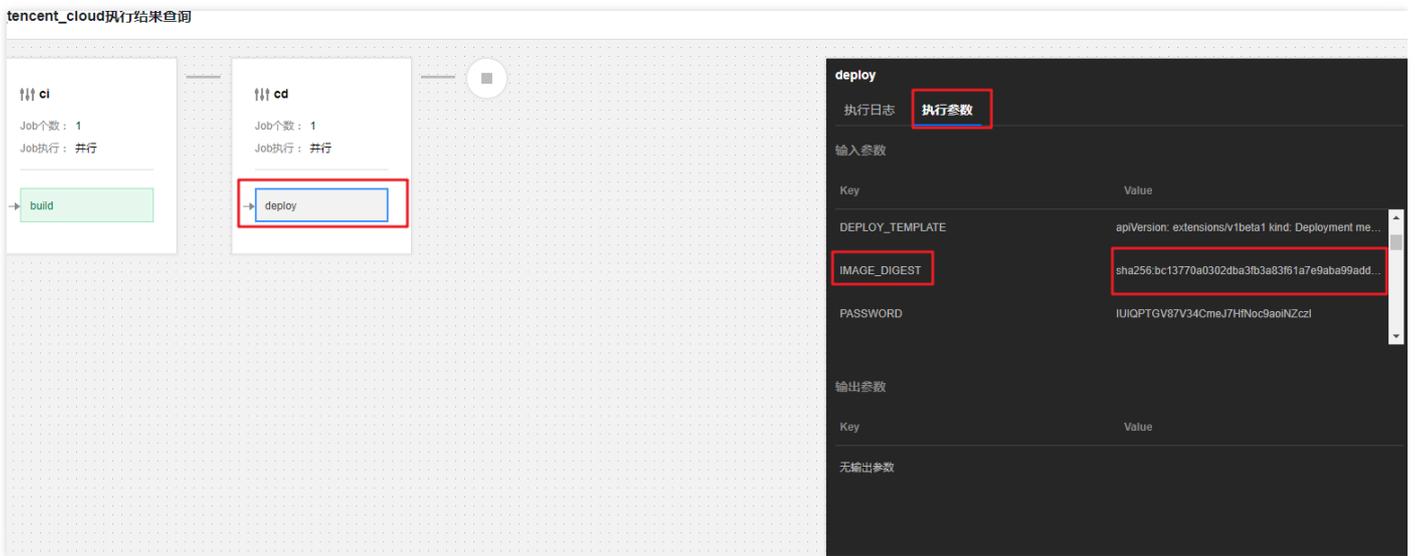
填写完毕后保存。

7. 推送代码到 <https://github.com/Kevin-Qiu/2Moons> 的 `Tencent_Cloud` 分支测试一下前面的 CICD 结果，单击查看执行历史可以查看执行明细。





可以在执行参数这里查看参数传递是否有误。



执行结果为成功。

8.切换到 TKE 查看一下部署情况。

容器服务 服务 广州 上海 上海金融 北京 成都 香港 新加坡 孟买 硅谷 重庆 曼谷 弗吉尼亚 莫斯科 所属集群 cjs-2jcz0la2 (temp) 所属集群空间

新建

名称	监控	日志	状态	运行实例数量	IP地址	负载均衡	标签(label)
moon			运行中	1/1个	172.19.255.103	未启用	qcloud-app:moon;
mariadb			运行中	1/1个	172.19.255.10	未启用	qcloud-app:mariadb;

应用中心 所有实例 所有容器 从 起始日期 00:00:00 至今 显示100条数据 自动刷新

```

40 [moon-5cb75bd96b-72rj6] [moon]Error: No files specified
41 [moon-5cb75bd96b-72rj6] [moon]Command: go-replace --mode=11ne --regex -s ^[\s]*myhostname[\s]*-* -r myhostname - moon-5cb75bd96b-72rj6
42 [moon-5cb75bd96b-72rj6] [moon][Thu Aug 02 13:14:15.059104 2018] [mpm_event:notice] [pid 67:tid 139666548574080] AH00489: Apache/2.4.18 (Ubuntu) OpenSSL/1.0.2g
configured -- resuming normal operations
43 [moon-5cb75bd96b-72rj6] [moon][Thu Aug 02 13:14:15.059697 2018] [core:notice] [pid 67:tid 139666548574080] AH00094: Command line: 'apache2 -D FOREGROUND -D
APACHE_LOCK_DIR'
44 [moon-5cb75bd96b-72rj6] [moon][02-Aug-2018 13:14:15] NOTICE: fpm is running, pid 66
45 [moon-5cb75bd96b-72rj6] [moon][02-Aug-2018 13:14:15] NOTICE: ready to handle connections
46 [moon-5cb75bd96b-72rj6] [moon][02-Aug-2018 13:14:15] NOTICE: systemd monitor interval set to 10000ms
47 [moon-5cb75bd96b-72rj6] [moon]2018-08-02 13:14:16,077 INFO success: syslogd entered RUNNING state, process has stayed up for > than 1 seconds (startsecs)
48 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix[163]: Postfix is running with backwards-compatible default settings
49 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix[163]: See http://www.postfix.org/COMPATIBILITY_README.html for details
50 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix[163]: To disable backwards compatibility use "postconf compatibility_level=2" and "postfix reload"
51 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/postfix-script[227]: warning: group or other writable: /usr/lib/postfix/.//libpostfix-global.so.1
52 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/postfix-script[228]: warning: group or other writable: /usr/lib/postfix/.//libpostfix-util.so.1
53 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/postfix-script[229]: warning: group or other writable: /usr/lib/postfix/.//sbin/lmp
54 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/postfix-script[230]: warning: group or other writable: /usr/lib/postfix/.//libpostfix-dns.so.1
55 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/postfix-script[231]: warning: group or other writable: /usr/lib/postfix/.//sbin/lmp
56 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/postfix-script[232]: warning: group or other writable: /usr/lib/postfix/.//libpostfix-master.so.1
57 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/postfix-script[233]: warning: group or other writable: /usr/lib/postfix/.//sbin/lmp
58 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/postfix-script[257]: starting the Postfix mail system
59 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] postfix/master[259]: daemon started -- version 3.1.0, configuration /etc/postfix
60 [moon-5cb75bd96b-72rj6] [moon][SYSLOG] CRON[363]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
    
```

Ingress是用于将外部HTTP(S)流量路由到服务 (service) 的规则集合。 [更多说明](#)

新建

ingress名称	状态	类型	VIP	后端服务
game	运行中	lb-gxhjdlyf 应用型负载均衡	203.195.128.116	http://game.tdemo.net-->moon:80

Deploy 已经顺利执行。

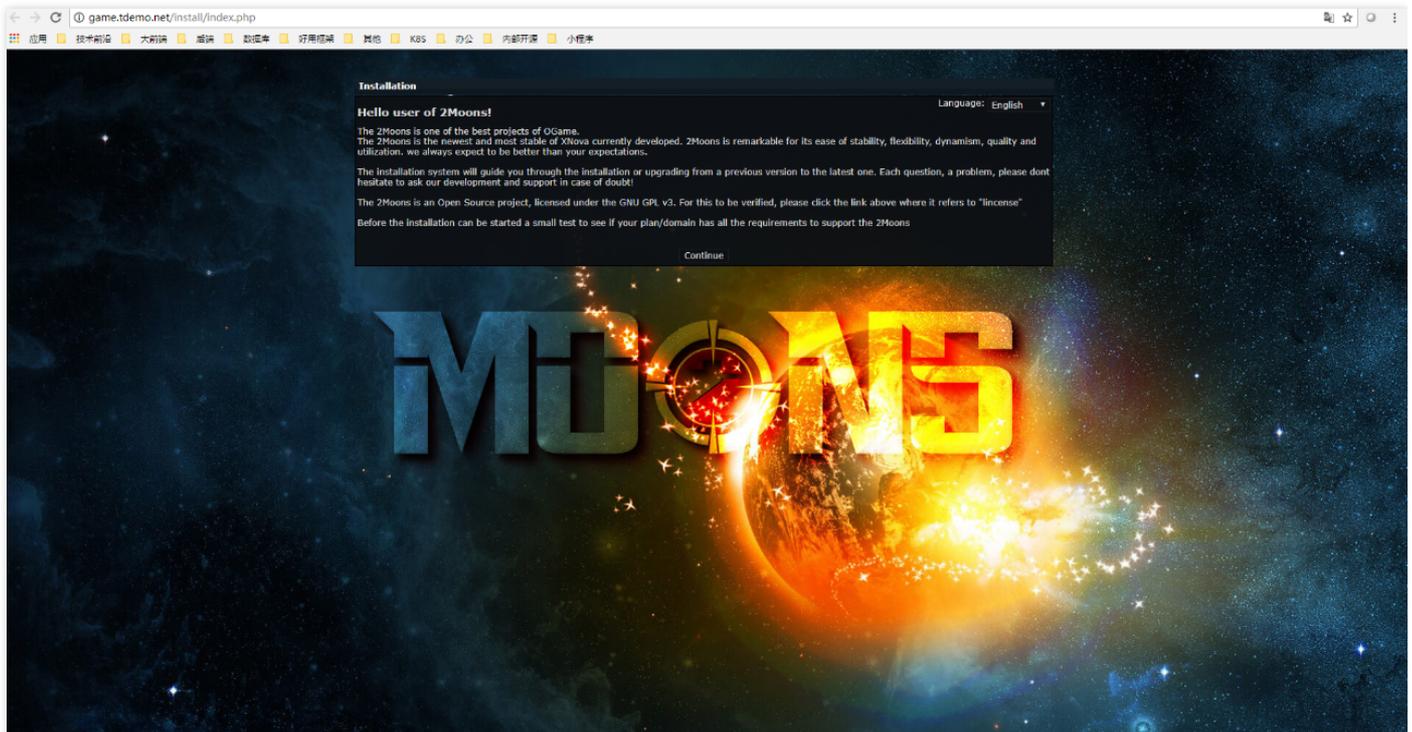
配置一下 game.tdemo.net 域名指向为 ingress 的外网 IP。

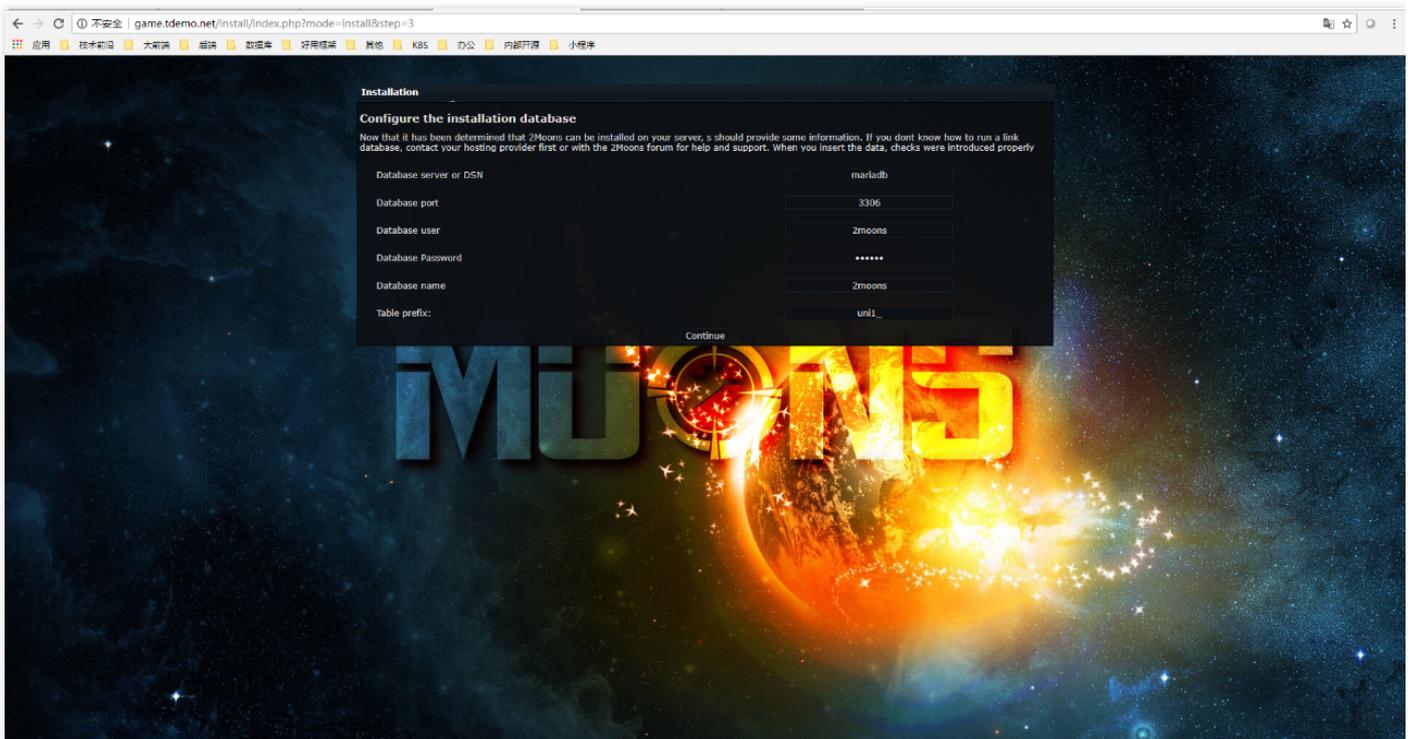
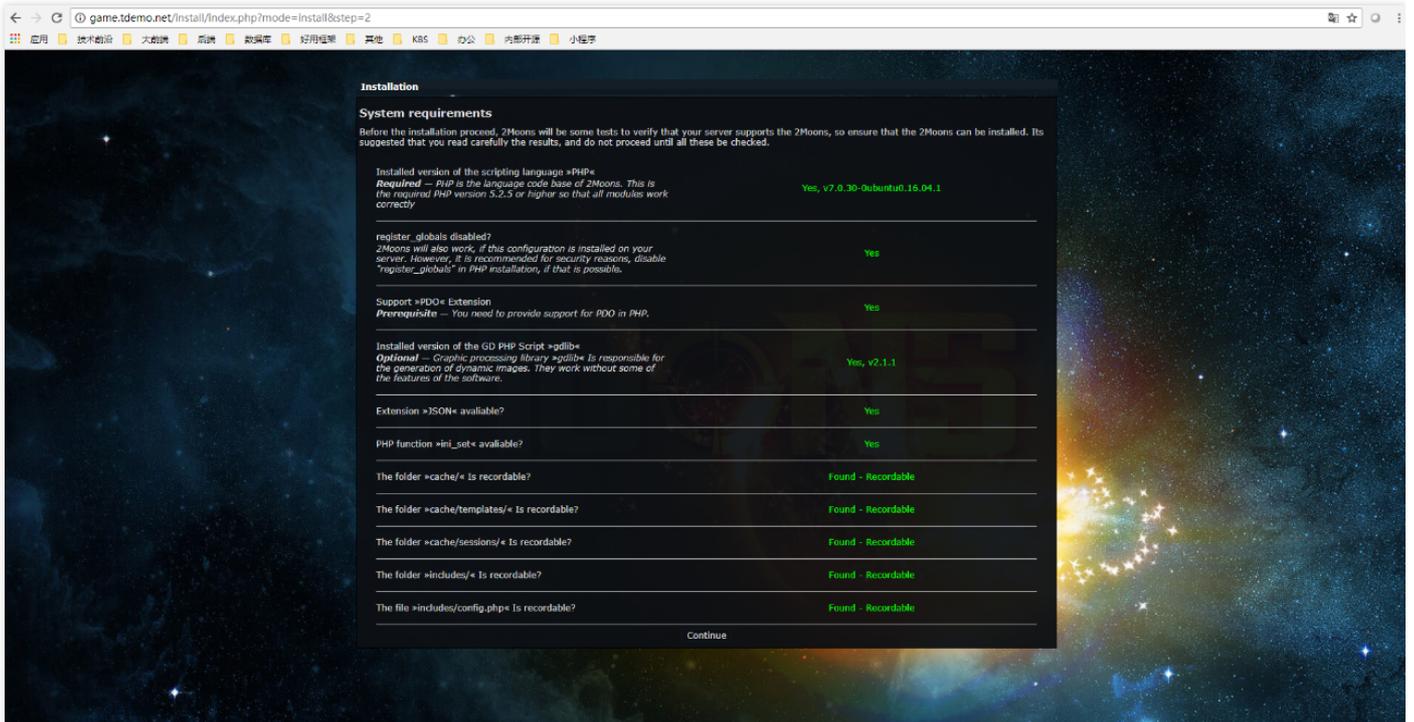
```
C:\Users\kaiqiu>ping game.tdemo.net

正在 Ping game.tdemo.net [203.195.128.116] 具有 32 字节的数据:
来自 203.195.128.116 的回复: 字节=32 时间=11ms TTL=49
来自 203.195.128.116 的回复: 字节=32 时间=10ms TTL=49
来自 203.195.128.116 的回复: 字节=32 时间=10ms TTL=49
来自 203.195.128.116 的回复: 字节=32 时间=11ms TTL=49

203.195.128.116 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 10ms, 最长 = 11ms, 平均 = 10ms

C:\Users\kaiqiu>
```





游戏到这里就部署好了。